OPERATIONS RESEARCH CENTER

working paper

**MASSACHUSETTS INSTITUTE
OF TECHNOLOGY**

Some Lattice-Theoretic Tools for
Network Reliability Analysis

by

Daniel Bienstock

OR 138-85

August 1985

This research was carried out at MIT and was partially funded by
grant NSF-ECS-8316224.

* Daniel Bienstock is now an Assistant Professor at the Graduate
School of Industrial Administration, Carnegie-Mellon University

Abstract

We present lattice-theoretic algorithms for computing connectivity parameters in graphs whose arcs are probabilistically erased. We describe some applications and give computational bounds.

Some Lattice-Theoretic Tools for Network Reliability Analysis

1. Introduction

Network reliability is the field of study that concerns graphs whose arcs and nodes are erased probabilistically. The analysis of such graphs arises in the design and operation of communication networks that are subject to damage.

There is a large amount of literature on network reliability problems. The vast majority of this work has been concentrated on the following computational problem: given a graph whose nodes and arcs fail independently, and a subset K of its nodes, evaluate the probability that the operative nodes of K remain connected.

In this paper we present an interesting lattice-theoretic identity concerning the connectivity of probabilistic graphs, which is valid for arbitrary failure distributions. This identity can be applied to compute relevant reliability parameters in the case of independent arc failures. We compare the computational requirements of the resulting algorithm to existing methods. Some further extensions and applications are outlined.

2. Some definitions and preliminaries

A state of a probabilistic network is a specification of the operating status of all its elements (nodes and arcs), that is a description of whether each element is ON or OFF. In the context of this paper, only arcs will fail, and thus states refer only to arcs.

An event is a collection of states. Let H be a graph with a set $E = \{1, 2, \dots, w\}$ of distinguished nodes. Consider an arbitrary partition $x = (B_1, B_2, \dots, B_j)$ of E , where the B_i 's are the blocks of x . Let $P(x)$ denote the probability of the following event: H is partitioned into j connected components $C_1, \dots, C_j$ such that component C_i , for $1 \leq i \leq j$, contains block B_i . An event of this form will be called a bond event. The states that make up a bond event will be called bond states. Given a partition x of E , $P(x)$ will be called a bond probability of H with respect to E , or bond probability for short.

Our objective is to compute the vector P of all bond probabilities of H w.r.t. E (where we order the partitions of E in some arbitrary fashion). The number of entries of P is equal to the w^{th} Bell number $B(w)$, that is the number of partitions of a w -element set. We will describe the asymptotic behavior of $B(w)$ at the end of this section.

The computation of bond probabilities is relevant in that it makes up the recursive step in two (closely related) algorithms [Rosenthal, 1977], [Fratta and Montanari, 1976] for computing reliability measures of graphs whose arcs fail independently. We will briefly sketch the common idea to these two algorithms later. We point out the following fact: whenever the graph H is dense and the set E is small, computing a bond probability should, in general, be much more difficult than computing the connectivity probability of H (or a graph similar to H). Intuitively, in order to show that a graph is connected, we only need to produce a spanning tree. On the other hand, in order to guarantee that a bond state occurs, we will need to produce several trees, and in addition, prove that the desired family of cuts is realized. Whenever H

is dense, we will be forced to carry out a large amount of enumeration. In this paper we will reduce the computation of bond probabilities to computing connectivity probabilities. In the case of independent arc failures, whenever H is dense and E has few nodes compared to the total number of nodes, our approach will be superior to the Rosenthal and Fratta & Montanari method.

Szekeres (1953), among others, proved the following result: if r is given by $re^r = w$, then

$$B(w) \sim \frac{1}{\sqrt{r}} e^{w(r + \frac{1}{r} - 1) - 1}.$$

Some estimations will show that

$$r = \log w - \log \log w + \frac{\log \log w}{\log w} + O\left(\left[\frac{\log \log w}{\log w}\right]^2\right),$$

from which we conclude that $B(w)$ is roughly exponential in $w \log w$. One can, in addition, show that $B(w) = o(w!)$. Nevertheless, $B(w)$ increases very rapidly with w (even for small w).

3. An identity for bond probabilities

In this section we describe the main result of this paper, a lattice-theoretic identity satisfied by the bond probabilities of a graph. As in previous sections, we are given a graph H that contains a distinguished subset E of w nodes. The arcs of H are erased according to some probability distribution, and we want to compute the vector P of bond probabilities of H w.r.t. E .

Given a partition $y = [B_1, \dots, B_k]$ of E , we will form graph $H[y]$ from H by collapsing all nodes in block B_i , for $1 \leq i \leq k$, into a single node (we keep all self-loops). Let $R[y]$ be the connectivity probability of $H[y]$ (where the arcs of $H[y]$ fail with the same distribution as in H), and R the vector of entries $R[y]$, one per each partition y of E . We will next show that there is a linear relationship between R and P .

Consider first the case $w = 2$. Then $H[1,2] = H$, and thus $R[1,2] = P[12] = \text{Prob}(H \text{ remains connected})$. We claim that

$$R[12] = P(12) + P(1,2). \quad (1)$$

In order to see why (1) holds, consider any state s counted in computing $R[12]$. State s implies that there is a path from every node in H to either 1 or 2. Thus state s will be counted in either $P(12)$ or $P(1,2)$ (but not in both since the corresponding bond events are disjoint). Similarly, every state counted in $P(12)$ or $P(1,2)$ will be counted in $R[12]$.

Consequently, we have that

$$R = A(2) P,$$

where $A(2)$ is the matrix

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

Let us briefly consider the case $w = 3$. Using the same arguments as for proving (1), we can show that, for example,

$$R[1,23] = P(123) + P(12,3) + P(13,2). \quad (2)$$

Notice that $P(1,23)$ does not appear in the right hand side. The reason for this fact is that each state s counted in $P(1,23)$ implies that there is a cut separating 1 from 2 and 3. Thus state s could not be counted in $R[1,23]$. Similarly, $P[1,2,3]$ is not counted.

Figure 1 - Example of identity (2).

In fact, if we order the partitions of $\{1,2,3\}$ as $/123/$, $/1,23/$, $/12,3/$, $/13,2/$ and $/1,2,3/$ (where $/$ stands for either $[$ or $($), we will have that

$$R = A(3)P, \quad (3)$$

where $A(3)$ is the matrix

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{matrix} .$$

In the general case, we can show that

$$R = A(w) P, \tag{4}$$

where $A(w)$ is a symmetric, 0-1 matrix described in Section 4. We will prove (4) in Section 5. The most relevant property of $A(w)$ is that it is always invertible (refer to Section 4 for more details). Consequently, in order to find the bond probabilities we can first compute the connectivity probabilities, and then solve the linear system (4). Suppose that H has n nodes and m arcs, and let $C(N,M)$ be the complexity of the algorithm used to compute the connectivity probability of a graph with N nodes and M arcs. As shown in the next section, we can compute the matrix $A(w)$ in $O(wB^2(w))$ time. It follows that we can compute the vector P in time

$$O(C(n,m) B(w) + wB^2(w) + B^3(w))$$

$$= O(C(n,m) B(w) + B^3(w)).$$

Notice that the above expression (modulo the factor $C(n,m)$) is polynomial in $B(w)$, the length of the vector we seek to compute.

A case of special interest occurs when the graph H remains invariant under arbitrary permutations of the set E . We will refer to this case as the symmetric case. In the symmetric case, system (4) is slightly "redundant." For example, if $w = 8$, then $R[124,356,7,8] = R[138,567,2,4]$ and $P(1,2345,678) = P(8,2467,135)$. In general, whenever two partitions x and y have the same number of blocks of each cardinality (in lattice-theoretic language, whenever x and y are of the same class), we will have that $R[x] = R[y]$ and $P(x) = P(y)$.

We can then simplify system (4) by applying to $A(w)$ the following operations:

- (1) select one row per partition class, and
- (2) replace each set of columns corresponding to a partition class by their sum.

For example, if we apply (1) and (2) to $A(3)$ we will obtain the matrix

$$\begin{array}{ccc} 1 & 3 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 0 \end{array} .$$

Let $A(2,w)$ be the resulting matrix. It follows that

$$R' = A(2,w) P', \tag{4'}$$

where the vectors R' and P' are obtained from R and P , respectively, by selecting one entry per each partition class.

As shown in the next section, matrix $A(2,w)$ is also invertible, and we can solve system (4') to compute the bond probabilities. What are the computational advantages of the symmetric case over the general case? The total number of partition classes of a w -element set, denoted by $p(w)$, satisfies

$$p(w) < e^{k\sqrt{w}}, \text{ where } k < 3,$$

and matrix $A(2,w)$ can be computed directly in time $O(wp^2(w)3^w)$ (see [Bienstock, 1985] for more details). Consequently, we can compute the vector P' in time

$$O(C(n,m) p(w) + wp^2(w) 3^w + p^3(w)) =$$

$$O(C(n,m) p(w) + wp^2(w) 3^w), \tag{5}$$

which is far preferable to the complexity of the general case (roughly, because $B(w)$ grows faster than exponentially in w). On the other hand, (5) is not polynomial in $p(w)$, and this difficulty arises because our method of computing $A(2,w)$ is not polynomial in $p(w)$. We have not been able to find a better method.

As a final remark in this section, we point out that matrices $A(w)$ and $A(2,w)$ are, in fact, completely independent of the particular graph

H (they only depend on w). Thus, systems (4) and (4') shed some light on the underlying combinational structure of probabilistic graphs.

5. The lattice of partitions

In this section we provide a brief introduction to lattice theory, as it pertains to the study of partitions. We will not give a general description of a lattice. For more background on this topic, the reader is referred to [Rota, 1964] or [Bienstock, 1985]. The reader is especially encouraged to consult Rota's paper.

The lattice of partitions L of a set S is the set of partitions of S , together with a partial order relation $\leq$, where we say $x \leq y$ if each block of x is a union of blocks of y . For example, if $|S| = 6$, $(123,456) \leq (12,3,4,56)$. The partition consisting of a single block that contains all of S is called 0 and satisfies $0 \leq x$ for all x in L . The partition in which every element of S forms its own block is called 1 and satisfies $x \leq 1$ for all x in L . We say that $y < x$ if $y \leq x$ but $y \neq x$.

The Möbius function $\mu: L \times L \rightarrow \mathbb{Z}$ is recursively defined as follows:

for all x , $\mu(x,x) = 1$; and for $x < y$,

$$\mu(x,y) = - \sum_{x \leq z < y} \mu(x,z).$$

The Möbius function of an arbitrary lattice is a construct of fundamental importance in combinatorial analysis. In this paper we will only touch on some of the properties of the Möbius function of a lattice of partitions. Two properties that we will not prove are the following. First, for all x in L , $\mu(0,x) \neq 0$. Also, consider a permutation $\pi: S \rightarrow S$. This permutation induces, in turn, a permutation on L , which we will also call π . Then, for all x and y ,

$$\mu(x,y) = \mu(\pi(x), \pi(y)).$$

The meet operator $\wedge: L \times L \rightarrow L$ is defined by $x \wedge y = \max\{w: w \leq x, w \leq y\}$ (this definition is valid for arbitrary lattices). In other words, the meet of x and y has the "smallest" blocks that are simultaneously partitioned by the blocks of x and the blocks of y . It is not difficult to see that the meet of x and y satisfies the following property: Consider a sequence $B_1, B_2, \dots, B_k$ of blocks of x and y , such that for i odd, B_i is a block of x ; and for i even, B_i is a block of y , and such that for $1 \leq i \leq k-1$, B_i and B_{i+1} share at least one element of S . Then $B_1 \cup B_2 \cup \dots \cup B_k$ will be contained in the same block of $x \wedge y$.

The preceding property can be interpreted as follows: to compute the meet of two partitions x and y , it is equivalent to find the connected components of a bipartite graph $G = (X, Y, L)$, where the nodes in X correspond to the blocks of x (and similarly with Y and y); and there is an edge joining two nodes whenever the corresponding blocks share at least one element of S . We do not prove this observation, but

point out that G has $O(|S|)$ nodes and $O(|S|)$ edges, and thus the connected components of G can be computed in $O(|S|)$ time.

If $x \wedge y \neq 0$ then G has more than one connected component (and viceversa). In that case we can write $S = S_1 \cup S_2$, where S_1 and S_2 are disjoint, and for $i=1,2$, S_i is a union of blocks of x and a union of blocks of y (i.e., x and y share a "cut" of S).

Armed with the meet operator and the Möbius function, we can describe the matrix $A(|S|)$ which appears in equation (4). $A(|S|)$ is defined by

$$a(x,y) = \begin{cases} 1, & \text{if } x \wedge y = 0 \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Definition (6) can be used in any lattice. A general property is that, whenever

$$\mu(0,x) \neq 0 \text{ for all } x, \quad (7)$$

then the matrix defined by (6) is invertible [Doubilet, 1972]. Since the lattice of partitions of any set S satisfies (7), we conclude that $A(|S|)$ is always invertible. Let B be the inverse of matrix (6). In terms of the Möbius function, B can be shown to have entries

$$b(x,y) = \sum_{t \geq x,y} \frac{\mu(x,t) \mu(y,t)}{\mu(0,t)}. \quad (8)$$

In the remaining part of this section we prove the invertibility of the matrix $A(2,d)$, introduced in the previous section, where $d = |S|$.

This matrix was obtained from $A(d)$, by (1) selecting from $A(d)$ one row corresponding to each partition class, and (2) replacing each set of columns corresponding to a partition class by their sum. The resulting matrix will be of size $p(d) \times p(d)$ (where, as before, $p(d)$ = no. of partition classes of a d -element set = no. of ways we can write d as a sum of positive integers), whereas $A(d)$ is of size $B(d) \times B(d)$ (where, as before, $B(d)$ = no. of partitions of a d -element set).

Set $m = p(d)$ and $M = B(d)$. We assume that we number the partition classes of S so that the j^{th} row of $A(2,d)$ corresponds to partition class j , denoted by C_j for short.

Theorem 1: Matrix $A(2,d)$ is invertible.

Proof: The proof is based on the explicit formula (8) for the inverse of $A(d)$. We will show that, for each unit m -vector e_i , there is an m -vector u^i satisfying $A(2,d)u^i = e^i$. This construction will prove invertibility.

Consider vector e^i , whose entry of 1 occurs in the coordinate corresponding to partition class C_i . Construct the 0-1 M -vector E^i with entries indexed by partitions of S , by setting

$$E^i(y) = \begin{cases} 1, & \text{if } y \in C_i \\ 0, & \text{otherwise.} \end{cases}$$

Since $A(d)$ is invertible, there exists an M -vector U^i satisfying $A(w)U^i = E^i$. Given x , formula (8) gives

$$U^i(x) = \sum_{y \in C_i} \sum_{t \geq x, y} \frac{\mu(x, t) (y, t)}{\mu(0, t)}.$$

Let z be any other partition of the same class as x . Then we can find a permutation π of S that maps x to z , by mapping blocks of each size into blocks of the same size.

Figure 2 - A permutation that maps x into z .

Then we will have

$$\begin{aligned}
 U^i(z) &= U^i(\pi(x)) = \sum_{\pi(y) \in C_i} \sum_{\pi(t) \geq \pi(x), \pi(y)} \frac{\mu(\pi(x), \pi(t)) \mu(\pi(y), \pi(t))}{\mu((0, \pi(t)))} \\
 &= U^i(x)
 \end{aligned}$$

since a permutation leaves the Möbius function and the class structure invariant.

Hence, in vector U^i , all entries corresponding to a given class C_k are equal to a common value $u^i(k)$. Thus, if x is any partition of class j (in particular, the one whose row was chosen in constructing $A(2,d)$), from $A(d)U^i = E^i$ we conclude that

$$E^i(x) = \sum_k (\# \text{ of } 1\text{'s in } A(d) \text{ along row } x, \text{ on columns of } C_k) \cdot u^i(k).$$

But the (j,k) entry of $A(2,d)$ is precisely the coefficient of $u^i(k)$ in this sum. Since $E^i(x) = e^i(j)$, we conclude that $e^i = A(2,d)u^i$, as desired.

6. Proof of lattice-theoretic identity for bond probabilities

In this section we will prove equation (4), $R = A(w)P$, where, as before, P = vector of bond probabilities of graph H w.r.t. a set of E of nodes, R = vector of connectivity probabilities, and $|E| = w$. The arcs of H are erased according to some probability distribution.

Theorem 2: $R = A(w)P$

Proof: Let x be a partition of E . Consider a state s counted in computing $R[x]$. Since state s implies that every node of H remains connected (via some path) to at least one node of E , it follows that s is a bond state and thus s is counted in some $P(y)$. We claim that $x \wedge y = 0$. Suppose not. Then (according to Section 5) we can write $E = E_1 \cup E_2$, with E_1 and E_2 disjoint, such that both E_1 and E_2 are the union of blocks of x and of y . This fact implies that (i) in state s , there is a cut separating all nodes of E_1 from those of E_2 , and (ii) to construct $H[x]$ we only identify nodes of E_1 to nodes of E_1 ; and

similarly with E_2 . Consequently, in state s graph $H[x]$ will be disconnected, and state s would not be counted in $R[x]$, a contradiction.

Similarly, if $x \Delta y = 0$, every state counted in $P(y)$ will be counted in $R[x]$. Therefore, $R[x] = \sum_{x \Delta y = 0} P(y)$, or

$$R = A(w)P,$$

as desired.

7. Computational considerations in the case of independent arc failures

In section 4 we saw that, in order to compute the bond probabilities of an n -node, m -arc graph H w.r.t. a subset E of w nodes, the lattice theoretic approach will have complexity

$$O(C(n,m)B(w) + B^3(w))$$

in the general case, and

$$O(C(n,m) p(w) + wp(w)3^w)$$

in the symmetric case, where $C(n,m)$ = complexity of the algorithm used to compute the connectivity probability of an n -node, m -arc graph under the given failure distribution. If the arcs of H fail independently, we can use Buzacott's algorithm [Buzacott, 1980] to compute the connectivity probabilities, in which case $C(n,m) = 3^n$.

If w is very small there is a more efficient implementation of the lattice-theoretic results. The key is that all the graphs $H[y]$ are very closely related to each other, hence, computing all $B(w)$ parameters $R[y]$ separately will repeat a large amount of enumeration. This wasted effort can be avoided as follows: we add to graph H a complete graph on E , where the reliability of the arc joining nodes i and j will be a variable called q_{ij} . Call the resulting graph H' . We run Buzacott's algorithm symbolically on graph H' . The reliability of H' will be a certain polynomial $r(q)$ of degree $w(w-1)/2$ in which each q_{ij} appears with degree 1. In order to compute the $R[y]$ parameters we simulate each partition y by setting the q_{ij} 's appropriately in $r(q)$. For instance, if $w = 3$ and we want to compute $R[1, 23]$, we would set $q_{12} = q_{13} = 0$, and $q_{23} = 1$.

When we generate a graph $H[y]$ we may create self loops (whenever two nodes in a block of y are adjacent in H). The state of such loops becomes irrelevant when computing $R[y]$, and consequently such loops may be deleted. We did not delete these in our proof of $R = A(w)P$ to simplify the exposition.

Finally, in a practical setting one may substitute Buzacott's algorithm for any other method that is available (or deemed more efficient). We have utilized Buzacott's algorithm in our discussion to obtain the best computational bounds.

8. Comparison with Rosenthal's algorithm

In the case of independent are failures, the very similar algorithms of Rosenthal (1977) and Fratta and Montanari (1976) can be

used to compute bond probabilities. In this section we describe the basic idea common to both algorithms, as it appears in Rosenthal's. We will refer to this idea as RFM.

Suppose we are trying to compute the bond probabilities of a graph H w.r.t. a set $E = \{1, 2, \dots, w\}$ of nodes. Consider a node cut that splits H into subgraphs H_1 and H_2 . Let the node cut consist of nodes $i_1, \dots, i_k$ and assume, without loss of generality, that H_1 contains the nodes $1, 2, \dots, r$ whereas H_2 contains the nodes $r+1, \dots, w$. Define $E_1 = \{i_1, \dots, i_k, 1, \dots, r\}$ and $E_2 = \{i_1, \dots, i_k, r+1, \dots, w\}$. We (recursively) compute the bond probabilities of H_j w.r.t. E_j , for $j=1, 2$. We can now compute the bond probabilities of H w.r.t. E : suppose that bond events e_1 and e_2 occur in H_1 and H_2 respectively. Then either

- (A) A subset of the nodes of H will remain isolated from E , or
- (B) Some bond event of H w.r.t E will occur. We indicate this event by $e_1 * e_2$ (there is a close relationship between the $*$ and the $\wedge$ operators). In general, for any bond event e of H w.r.t. E there will be several pairs of bond events e_1 and e_2 such that $e = e_1 * e_2$.

Figure 3. Case (B) with $w=k=3$, $r=1$

Consequently, we need only enumerate all pairs of bond events e_1 and e_2 ; while keeping a running total for each bond probability of H w.r.t. E .

Algorithm RFM suffers from several problems:

1. The precise sequence of cuts to be taken is not specified. In general, one would want to keep the largest cut as small as possible.
2. The algorithm becomes inefficient when applied to very dense graphs. In the worst case, H is a complete graph, and we cannot split a complete graph with a node cut. In general, if H has n nodes and $\sim \frac{n^2}{2} - cn$ arcs, then H will contain a clique with $\Omega(\frac{n}{c})$ nodes (this result is not difficult to show). Hence, at some recursive step, RFM will run into difficulties.

Let us focus on point (2). Suppose that set E has w nodes, H has n nodes and $\frac{n^2}{2} - cn$ arcs. The lattice-theoretic approach will have complexity $O(B(w)e^n + B^3(w))$, whereas RFM will have complexity, roughly, $\Omega(B(\frac{n}{c}))$. Thus, if $w \ll n$, the lattice-theoretic approach is much superior to RFM.

What can we say about RFM in the case when H is sparse, indeed, if we can provide a recursive decomposition of H into sparse subgraphs? An archetypal case is that of planar graphs. In fact, a careful implementation of RFM will have complexity that is essentially strictly exponential in $\sqrt{n}$ (far superior to the lattice-theoretic approach). This result will be described in depth in a forthcoming paper.

9. Some further remarks concerning the case of independent arc failures.

The lattice-theoretic approach is also inefficient if the graph H has some strong symmetric properties, for instance if H is complete or complete k -partite (for any constant k), and all of the arcs of H are equally reliable. In fact, one can compute the bond probabilities in time

$$O(g(n)p^2(w)),$$

where $g(\cdot)$ is a polynomial. The algorithms essentially are a generalization of Buzacott's. In this context, it is possible to view Buzacott's algorithm as performing enumeration over the zero element of a certain lattice. Our algorithms generalize Buzacott's to higher elements of the same lattice.

Finally, as we saw in previous sections, the relationship $R = A(w)P$ leads to an algorithm for computing bond probabilities that is certainly not polynomial in n or w . Nevertheless, the graph H could be a subgraph of a much larger graph that we are analyzing. Viewed in this way, our lattice-theoretic methods lead to polynomial-time reliability evaluation algorithms for certain special graphs with relevant asymptotic properties.

We will describe the above results in a future paper.

References

1. Bienstock, D., "Large-Scale Network Reliability," Ph.D. Thesis, MIT Operations Research Center (1985).
2. Buzacott, J. A., "A Recursive Algorithm for Finding Reliability Measures Related to the Connection of Nodes in a Graph," Networks 10 (1980), pp. 311-327.
3. Doubilet, P., "On the Foundations of Combinatorial Theory VII: Symmetric Functions through the Theory of Distribution and Occupancy," Studies in Appl. Math. 51 (1972), pp. 377-395.
4. Fratta, L. and Montanari, U., "Decomposition Techniques for Evaluating Network Reliability," IBM report RC-5923 (#25618) (1976).
5. Rosenthal, A., "Computing the Reliability of Complex Networks," SIAM J. of Appl. Math. 32 (1977), pp. 384-393.
6. Rota, G.-C., "On the Foundations of Combinatorial Theory 1: Theory of Mobius Functions," Zeit. fur Wahr. 2 (1964), pp. 340-368.
7. Szekeres, "Some Asymptotic Formulae in the Theory of Partitions," Quarterly S.M. Oxford 4 (1953), pp. 96-111.